Random Forest

The Power of Ensemble Learning

Ahmed BADI

ahmedbadi905@gmail.com

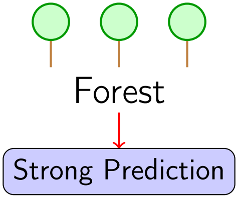
December 13, 2025

Outline

What is Random Forest?

The Wisdom of Crowds

- Instead of trusting one decision tree...
- Build many trees and let them vote!
- Like asking 10 friends for movie recommendations



Key Idea:

Combine multiple weak learners to create a strong learner

Why Random Forest?

Advantages:

- ✓ High accuracy
- ✓ Robust to overfitting
- ✓ Handles non-linear relationships
- ✓ No feature scaling needed
- ✓ Built-in feature importance
- ✓ Works for classification & regression

Applications:

- Medical diagnosis
- Credit risk assessment
- Fraud detection
- Customer segmentation
- Stock prediction
- Image classification

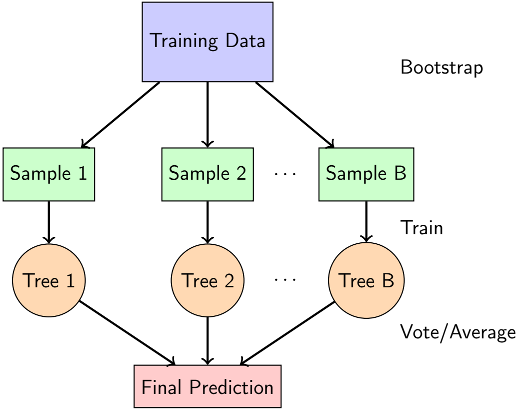
The Two Key Ideas

1. **Bootstrap Aggregating (Bagging)**
   - Create different training sets by random sampling with replacement
   - Train one tree on each bootstrap sample
   - Average predictions (regression) or vote (classification)
2. **Random Feature Selection**
   - At each split, randomly select m features
   - Choose best split among only these m features
   - Typical: $m = \sqrt{p}$ for classification, $m = p/3$ for regression

Result

Trees are diverse $\rightarrow$ Errors cancel out $\rightarrow$ Better predictions!

The Random Forest Process



Bootstrap Sampling

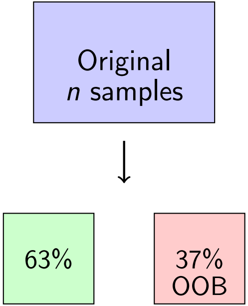
How it works:

- Sample n examples with replacement
- Some examples appear multiple times
- Some don't appear at all

Mathematics:

$$P(\text{selected}) = 1 - \left(1 - \frac{1}{n}\right)^n$$
$$\approx 1 - \frac{1}{e} \approx 0.632$$

About 63% of data in each bootstrap sample!



Out-of-Bag (OOB) Error

Free Validation!

Each tree sees only 63% of data. The remaining 37% (out-of-bag) can be used for validation.

How OOB works:

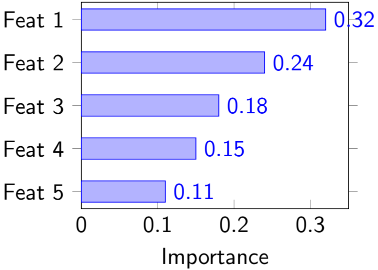
1. For each sample, find trees that didn't use it
2. Let these trees predict the sample
3. Compare with true label
4. Average error across all samples

Advantage

OOB error $\approx$ Test error. No need for separate validation set!

Two Methods:

1. **Mean Decrease Impurity**
   - Sum impurity reduction when splitting on feature
   - Fast but can be biased
2. **Permutation Importance**
   - Shuffle feature and measure accuracy drop
   - More reliable



Use case: Identify which variables drive predictions

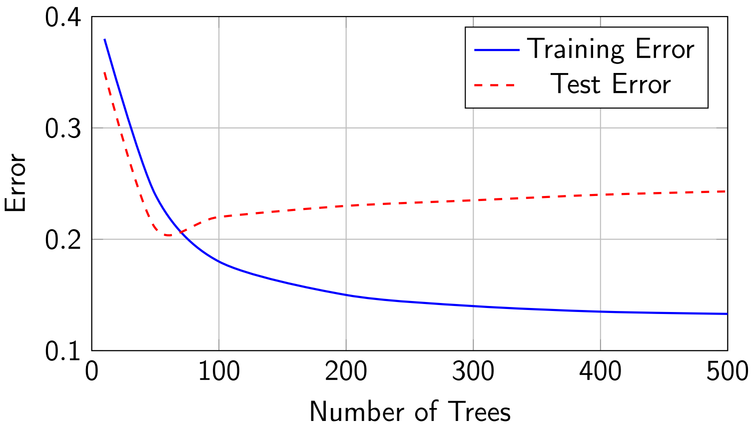
Key Hyperparameters

Parameter	What it does	Typical values
n_estimators	Number of trees	100-500
max_features	Features per split	$\sqrt{p}$ (classif), $p/3$ (regr)
max_depth	Tree depth limit	None (unlimited)
min_samples_split	Min samples to split	2-10
min_samples_leaf	Min samples in leaf	1-5
bootstrap	Use bootstrap?	True

Tuning Strategy

1. Start with defaults (they work well!)
2. Increase n_estimators (more is better)
3. Tune max_features if needed
4. Use OOB error to monitor performance

Number of Trees vs Error



Key insight: Test error stabilizes after 200 trees. More trees = more stable, but diminishing returns.

Random Forest vs Other Algorithms

Algorithm	Pros vs RF	Cons vs RF
Decision Tree	Faster, interpretable	Much less accurate, overfits
Gradient Boosting	Slightly higher accuracy	Slower, more tuning needed
Logistic Regression	Fast, simple, probabilistic	Can't handle non-linear data
Neural Networks	Best for complex patterns	Needs lots of data, black box

When to use Random Forest?

- Need high accuracy with minimal tuning
- Have mixed data types (numerical + categorical)
- Want feature importance
- Need robust, reliable predictions

RF vs Gradient Boosting

Random Forest:

- \+ Trains trees in parallel
- \+ Faster training
- \+ More robust (less overfitting)
- \+ Easier to use
- \+ Has OOB error
- – Slightly lower accuracy

Gradient Boosting:

- \+ Often 1-2% more accurate
- \+ More flexible
- – Sequential training (slower)
- – More prone to overfitting
- – More hyperparameters
- – Requires careful tuning

Rule of thumb

Start with RF. Try boosting if you need that extra accuracy and have time for tuning.

Real-World Applications

Healthcare:

- Disease prediction
- Drug discovery
- Patient risk assessment
- Medical image analysis

Finance:

- Credit scoring
- Fraud detection
- Stock price prediction
- Risk management

E-commerce:

- Recommendation systems
- Customer churn prediction
- Demand forecasting
- Price optimization

Other domains:

- Environmental monitoring
- Manufacturing quality control
- Cybersecurity
- Agriculture

Advantages

- ✓ **High accuracy** - Top performer
- ✓ **Robust** - Resistant to overfitting
- ✓ **Versatile** - Classification & regression
- ✓ **No preprocessing** - No scaling needed
- ✓ **Feature importance** - Built-in
- ✓ **Handles missing data** - Naturally
- ✓ **Parallelizable** - Fast on multi-core
- ✓ **OOB validation** - Free error estimate
- ✓ **Few hyperparameters** - Easy to tune
- ✓ **Works well** - With default settings

Limitations

Disadvantages:

- × Less interpretable than single tree
- × Larger model size (memory)
- × Slower predictions than simple models
- × Can't extrapolate (regression)
- × Biased with high-cardinality features

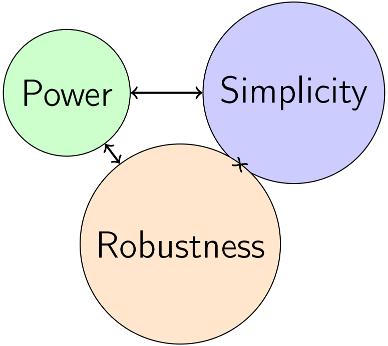
When NOT to use:

- Very small datasets ($<$ 100 samples)
- Need maximum interpretability
- Prediction speed is critical
- Working with text/images (use DL)
- Linear relationships dominate

Key Takeaways

1. **Random Forest = Bagging + Random Features**
   - Build diverse trees through randomization
   - Aggregate predictions to reduce variance
2. **Excellent out-of-the-box performance**
   - Often works well with default parameters
   - Less tuning than other algorithms
3. **Best practices:**
   - Use 100+ trees
   - Monitor OOB error
   - Check feature importance
   - Start here, optimize later

Random Forest



"The best algorithm is the one you can understand, implement, and trust."

Random Forest delivers on all three.

Thank You!

Questions?

Ahmed BADI
ahmedbadi905@gmail.com
linkedin.com/in/badi-ahmed